98-Comp-A4

Program Design and Data Structures

3 Hours Duration

Notes:

1. If doubt exists as to the interpretation of a question, the candidate is urged to submit with the answer paper a clear statement of any assumptions made.

2. No calculator permitted. This is a Closed book exam.

3. Answer any five of the eight questions.

4. Any five questions constitute a complete paper. Only the first five questions as they appear in your answer book will be marked.

5. For questions that ask the candidate to write a program, **pseudocode** or any high-level language (e.g. **C** or **C++**) is acceptable unless otherwise specified. In all cases, marking will emphasize the operation of the program and not syntactic details.

6. All questions have equal weight.

Question 1. *Programming.*

(a) Floyd's triangle lists integers in a right triangle that is aligned to the left. The first row is simply the integer 1. Successive lines start with the next integer followed by successive integers listing one more integer than the line above. Here is what the triangle looks for 5 lines:

```
 1
 2  3
 4  5  6
 7  8  9 10
11 12 13 14 15
```

Write a program to read the number of rows `n` from the user and then generate and output the first `n` lines of Floyd's triangle.

(b) Three desperados stole a shipment of gold bars late one night. After escaping to their hideout, they resolved to divide the booty in the morning and went to bed. However, as soon as one of the bandits heard the others snoring, he split the stolen gold into three equal piles, and found one bar left over. He buried one of the three piles under a tree, along with the extra bar, and left the rest of the gold. Then he went to sleep, sure that he had protected his interest in the treasure. Naturally, the other two outlaws were no more honest than the first. Each in turn crept to the cache of gold, divided it three ways, and found one bar left over, which he kept along with "his third".

Soon came the morning and the final three-way division. Oddly enough, this division also left one bar remaining. The highwaymen fought over this bar, and in an unprecedented three-way draw, shot each other dead.

Write a program to find how many bars could have been in the original loot. Assume that the loot contained no more than 500 bars.

Hint: Write a subprogram to determine if the above divisions of the gold are possible for a given number of bars and then use it.

Question 2. *Programming.*

An integer is said to be "self-describing" if the following holds. When digit positions are labeled 0 to n-1 from left to right, the digit in each position is equal to the number of times the digit position appears in the number. Thus, the integer 2020 is a 4-digit self-describing integer. Position 0 has value 2 and there are two 0's in the number; position 1 has value 0 and there are no 1's in the number; position 2 has value 2 and there are two 2's; and position 3 has value 0 since there are zero 3's. These are also self-describing integers: 1210 and 3211000.

Write a program that prompts the user for a positive integer and responds by printing a message that indicates if the entered number is self-describing or not.

Hint: you may want to read the digits into an array.

Question 3. *Programming.*

A **magic square** is a square array of integers such that the sum of every row, the sum of every column and the sum of each of the two diagonals are all equal. This is an example of a 4x4 magic square.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

Write a program that reads the size of the square n and allocate an nxn two-dimensional array of integers. The program then reads n^2 integers that represent the rows of the square from top to bottom. The program finally determines whether or not the square is a magic square.

Question 4. *Object-Oriented Design.*

Vectors are used in many engineering applications. However, some languages do not have a "vector" data type, nor do they directly support vector operations.

Design and write a C++ class (call it `Vector`) for supporting vectors and their operations. Your class should allow for the declaration of a vector of a given size, without initialization of its elements. Your class should allow for the accessing (read & write) of elements of a vector. It should also allow for the addition, subtraction, multiplication, and printing of

vectors. These operations should produce an error message if the vectors involved in an operation are not of the same size.

Use C++ templates to support vectors of various types (only assume **`int`**, **`float`**, and **`double`** for simplicity). Overload the usual arithmetic operators to provide addition, subtraction, and multiplication of two vectors. Also overload the equality operator to allow equality comparison of two vectors.

You have freedom to select the exact syntax of some of the above operations. State any assumption you make clearly. Separate your class into a **`Vector.h`** header file and a **`Vector.cc`** implementation file.

Question 5. *Pointer-based Data Structures.*

(a) Consider the following definitions of a `node` structure and of a `stack` module.

```
typedef struct {
    int data;
    node *next;
} node;


#ifndef STACK_H
#define STACK_H

static node *top = NULL;

void make_empty(void);
int is_empty(void);
void push(int i);
int pop(void);

#end if
```

Write an implementation of the `stack` module using a linked list of `nodes`. Recall that a stack is a list of elements with insertions and deletion done at only one end of the list, called the "top" of the stack.

(b) Assume a singly-linked list of `node` structures, where a `node` structure is as defined in part (a) above. The head of the list is a variable `head`, which points to the first node in the list. Write a function that prints the linked list in *reverse* order. That is, the function should first print the data field of the last node on the list, followed by the data field of the previous node, and so on until it prints the data field of the first node, pointed to by `head`.

Question 6. *File I/O.*

A plain text file may contain 3 types of parentheses: round brackets: ( ), square brackets: [ ], and curly brackets: { }. Parentheses are *balanced* if every opening parenthesis is closed in the reverse order opened. Thus '([ ])' is balanced but '([ )]' is not. Write a program to prompt the user for a file name, open and read plain text from the file and determine if parentheses that appear in the text are balanced or not. The program should print a simple message that indicates if all parentheses are balanced or not.

Hint: You may want to use a stack. If you decide to use one, you need not show the code that implements it, just use it.

Question 7. *File I/O.*

The Securities and Exchange Commission (SEC) has in its possession three files on disk. One lists all employees of firms that rendered financial advice to a particular Wall Street company. The second file has the names of all individuals who traded heavily in that company's stock. The third consists of every name found in the personal Rolodex of a financial advisor, recently convicted of insider trading. Each file is in alphabetical order.

Write a program that helps the SEC search for illegal insider traders by finding and printing names common to all three files.

Assume that names in each file appear one per line, and are all in the same format of: last name, first name. However, the number of lines in each file is not known.

Question 8. *Algorithm Design and Sorting.*

A programmer is given an array A of random integers. The integers in the array are not in any sorted order. However, the array may contain duplicate integers. The programmer must create another array B that contains all the integers in A but without any duplicates. The integers in B need not be in any sorted order. The programmer is contemplating two ways to accomplish this task:

1. For each integer in array A, copy the integer into array B unless the item already exists in B.

2. Sort A using the Quicksort sorting algorithm. Now repeat part (a).

Given that the array contains a large number of integers, which of the above two methods will be faster? Justify your answer.